Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW	Docket Number (Optional) 1801270.00138US1	
	Application Number 10/749,052-Conf. #7395	Filed December 30, 2003
	First Named Inventor Ian G. BOLTON et al.	
	Art Unit 2192	Examiner E. B. Kiss

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

- [ ] applicant /inventor.
- [ ] assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)
- [x] attorney or agent of record. Registration number 42,478
- [ ] attorney or agent acting under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34. _____

Signature

Ronald R. Demsher
Typed or printed name

(617) 526-6000
Telephone number

Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

- [ ] *Total of 1 forms are submitted.

Claims 1-45 are pending in this application.

SUMMARY OF PROSECUTION HISTORY

On October 2, 2007, the Examiner issued a non-final Office Action rejecting claims 3-15, 18-30 and 33-35 under 35 U.S.C. §112, second paragraph, claims 31-45 under 35 U.S.C. §101, and claims 1, 2, 16, 17 and 32 under 35 U.S.C. §102(b).

In a response filed on March 28, 2008, the Applicant amended claims 1-4, 6, 7, 11, 16-19, 21, 22, 26, 31-45 to address the Examiner's rejections from the Office Action of October 2, 2007.

On August 4, 2008, the Examiner issued a Final Office Action withdrawing the previous rejections under 35 U.S.C. §112, second paragraph, and 35 U.S.C. §101 in view of the Applicant's amendments and arguments with respect to those rejections. The Examiner deemed the Applicant's remaining arguments moot in view of new grounds of rejections, including rejections of claims 1-12, 16-27 and 31-42 under 35 U.S.C. §102(b), and rejections of claims 14, 15, 29, 30, 44 and 45 under 35 U.S.C. §103(a). The Examiner deemed claims 13, 28 and 43 to be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

This August 4, 2008 Final Office Action is currently pending.

CURRENT STATE OF PROSECUTION

35 USC §102

At paragraph 6 of the pending Final Office Action, the Examine rejects claims 1-12, 16-27 and 31-42 under 35 U.S.C. §102(b) as being anticipated by Munehiro Takimoto and Kenichi Harada, "Partial Dead Code Elimination Using Extended Value Graph," 1999, Springer-Verlag, Lecture Notes in Computer Science, vol. 1694, proc. of the 6th Int'l Symposium on Static Analysis, pp. 179-193 (PDE1999). The Applicant traverses these rejections.

The Applicant submits that PDE1999 does not disclose "A method of performing dynamic binary translation to convert subject program code executable on a subject computing architecture into target code executed by a target computing system" as recited in claim 1. The pending Office Action refers to Section 2 "Preliminaries" of PDE1999 on page 150. As shown in Figure 8 and the

related text, PDE1999 discloses that the nodes 1, 2, 3 represent "basic blocks of statements" and that edges represent "the non-deterministic branching structure" within the directed flow graph G. These statements are classified into the three groups, namely assignment statements, empty statements and relevant statements. However, Section 2 of PDE1999 does not explicitly disclose the step of "grouping together a plurality of basic blocks of the subject program code to form a group block" as recited in claim 1.

Further, Section 2 of PDE1999 does not disclose the step of "decoding the plurality of basic blocks of the subject program code in the group block" as recited in claim 1. Further still, PDE 1999 does not disclose "generating an intermediate representation including one or more nodes representing a register definition by the subject program code" as recited in claim 1.

The Examiner refers to Section 4.2 "Sinking-Sinking Effects" on page 152 of PDE 1999. However, Section 4.2 of PDE 1999 does not disclose the step of "performing a partial dead code elimination optimization on said intermediate representation [that] transverses the intermediate representation to create a liveness analysis indicating when the register definition represented by the one or more nodes is a partially dead register definition...". That is, PDE1999 simply does not discuss such "register definitions" as recited in claim 1.

Finally, Section 6 "Complexity" on pages 154-157 of PDE1999 cited in the Office Action does not disclose the steps of "generating target code from said optimized intermediate representation" or "executing said target code on said target computing system" as recited in claim 1.

In summary, PDE1999 cited in the Office Action fails to disclose one or more limitations in claim 1 and thus the rejection under 35 USC §102(b) is improper and should be withdrawn. Similar considerations apply to each of the rejected dependent claims, which are likewise not anticipated by PDE1999, at least because they depend from an allowable base claim. Similar considerations also apply concerning the computer-readable storage medium of claims 16-30 and the computer apparatus of claims 31-45.

35 USC §103(a)

At paragraph 8 of the pending Office Action, the Examiner rejects claims 14, 15, 29, 30, 44 and 45 under 35 U.S.C. §103(a) as being unpatentable over PDE1999, in view of Alfred V. Aho et al., "Compilers: Principles, Techniques and Tools," 1988, Addison-Wesley, pp. 554-555 (Aho1988). Aho1988 does not supply that which is missing from PDE1999. As set forth in the response filed on March 28, 2008, Aho1988 only discloses methods of static compilation from a high-level source language to a machine-readable target code. See for example the statements on page 587. By contrast, claim 1 recites "A *method of performing dynamic binary translation to convert subject program code executable on a subject computing architecture into target code executed by a target computing system*". Thus, the Applicant submits that the methods of static compilation disclosed by Aho1988 do not anticipate the method of performing dynamic binary translation as required by claim 1.

Claim 1 recites:

a.) "*grouping together a plurality of basic blocks of the subject program code to form a group block*";

b.) "decoding the plurality of basic blocks of the subject program code in the group block";

c.) "generating an intermediate representation from said plurality of basic blocks of the subject program code in the group block, wherein the intermediate representation comprises nodes and links arranged as a directed acyclic graph representing expressions, calculations and operations performed by the subject program code, including one or more nodes representing a register definition by the subject program code";

d.) "performing a partial dead code elimination optimization on said intermediate representation to generate an optimized intermediate representation, wherein the partial dead code elimination optimization traverses the intermediate representation to create a liveness analysis indicating when the register definition represented by the one or more nodes is a partially dead register definition which is live in one path and dead in another path through the group block"; and

e.) "generating target code from said optimized intermediate representation; and executing said target code on said target computing system.

As to feature a.) above, Aho1988 discloses, on page 591, that "in the code optimizer, programs are represented as flow graphs, in which edges indicate the flow of control and nodes represent basic blocks." Aho1988 discloses that "a basic block is a sequence of consecutive statements in which flow of control enters at the beginning and leaves at the end without halt or possibility of branching except at the end" (Aho1988, page 528). Therefore, the Applicant respectfully submits that Aho1988 only discloses that code is optimized using basic blocks having one entry point, one exit point and in which a series of consecutive statements are defined. The Applicant respectfully submits that Aho1988 contains no disclosure or suggestion relating to this step of grouping together plural basic blocks to form a group block as in feature a.) of claim 1.

As to feature b.), it follows that, because Aho1988 does not anticipate the use of group blocks, the step of "decoding the plurality of basic blocks of the subject program code in the group block" is also not anticipated by Aho1988.

As to feature c.), the applicant respectfully submits that Aho1988 does not disclose or even suggest generating an intermediate representation from said plurality of basic blocks of the subject program code in the group block, not least because Aho1988 does not disclose such group blocks. Furthermore, the Examiner argues that the intermediate code disclosed on pages 598-590 and Figure 10.4 of Aho1988 is equivalent to the intermediate representation of claim 1. However, it is clear from Fig. 10.4 that the intermediate representation is a series of statements representing the code fragment of Fig 10.2 of Aho1988. By contrast, feature c.) of claim 1 recites that the intermediate representation is "a directed acyclic graph". Further, the directed acyclic graph of claim 1 represents "expressions, calculations and operations performed by the subject program code" and includes "one or more nodes representing a register definition by the subject program code". That is, the directed acyclic graph represents how the subject program code uses the hardware resources of the subject computing architecture. Thus, the applicant respectfully submits that feature c.) of claim 1 is not anticipated by Aho1988.

As to feature d.), it also follows that Aho1988 does not disclose or suggest the step of "performing a partial dead code elimination optimization on said intermediate representation to generate an optimized intermediate representation" as recited in claim 1. Here, the dead code elimination of Aho1988 is not equivalent to the partial dead code elimination required by the method claim 1. Dead code elimination involves identifying code that is "live at a point in a program" or "dead at that point" (Aho1988, page 595). No other state is considered. By contrast, partial dead code optimization involves identifying code that is dead, code that is alive, and code that is partially dead. This partially dead code is dead in one branch of its destinations but live in the other of its branch destinations. It follows, therefore, that creating "a liveness analysis indicating when the register definition represented by the one or more nodes is a partially dead register definition which is live in one path and dead in another path through the group block" as recited in claim 1 is not disclosed or suggested by Aho1988.

Finally, as to feature e.), it follows that these steps are also not disclosed by Aho1988.

Similar arguments to those set forth above for claim 1 also apply to the remaining independent claims. Since Aho1988 does not supply that which is missing from PDE1999, the rejections of claims 14, 15, 29, 30, 44 and 45 under 35 U.S.C. §103(a) are improper and should be withdrawn. These claims all depend from allowable base claims.

In view of at least the arguments set forth above, the Applicant believes all pending claims 1-45 should be allowable, and the application is in condition for allowance.